

SPECIFICATION

Electronic Version 1.2.8

Stylesheet Version 1.0

A Protocol for a Power Supply Unit Controller

Cross Reference to Related Applications

This application claims priority from copending application number 60/204,577 filed May 16, 2000, which is hereby incorporated herein by reference.

Background of Invention

- [0001] The present invention relates to a protocol for a power supply unit (PSU) controller.
- [0002] In a rack enclosure comprising a backplane across which a plurality of devices, for example, storage disks communicate across a bus, for example, a SCSI bus, one or more power supply units are frequently used to supply power in the form of one or more voltage supply rails to the backplane for use by the devices. With the increasing complexity of configuration of rack enclosures, their continued operation can become more critical and so it becomes important not alone to diagnose faults when they have occurred but also to determine when faults may occur and what their cause might be.
- [0003] The present invention seeks to mitigate the problems of detecting and diagnosing problems which may occur in a rack enclosure power supply unit.

Summary of Invention

- [0004] According to the present invention there is provided a power supply unit controller for a rack enclosure in which a plurality of devices communicate via a backplane, said controller comprising:

[0005] means for reading at least one signal indicative of an output supply level being provided to said backplane by a power supply unit associated with said power supply unit controller;

[0006] memory for storing at least one value associated with a respective one of the at least one signal, at least one scaling value associated with a respective one of the at least one signal and dependent on said power supply unit, and a power supply unit serial number;

[0007] communicating means, responsive to a request from one of said devices, for a returning a state of said associated power supply unit to said requesting device, said state including a combination of:

[0008] a summary of the current status of the power supply unit,

[0009] said at least one value,

[0010] said at least one scaling value, and

[0011] said power supply unit serial number,

[0012] according to said device request.

[0013] The power supply unit controller of the present invention stores scaling factors dependent on the supply levels supplied by the power supply unit associated with the controller. These can be retrieved by, for example, a higher level processor monitoring environmental conditions in an entire rack enclosure. Using the scaling factor with its associated value read from the power supply unit by the power supply unit controller, the software in the higher level processor and the power supply unit controller itself can be made independent of the power supply unit itself. Furthermore, storing the power supply unit serial number in the controller enables messages identifying the specify power supply unit to be relayed to an operator, who can then correlate the message with a label on the power supply unit indicating the power supply unit serial number. Clearly not all information is required by a higher level processor in response to every request, and so the request can be formatted to condition the amount of information returned by the

power supply unit controller.

- [0014] According to a second aspect of the invention there is provided a rack enclosure including a backplane, at least one power supply unit connected to and adapted to supply power to said backplane, each associated with a respective power supply unit controller according to the invention, and a plurality of devices receiving power from said backplane, at least one of said devices adapted to communicate with the at least one power supply unit controller.

Brief Description of Drawings

- [0015] An embodiment of the invention will now be described with reference to the accompanying drawings, in which:
- [0016] Figure 1 is a block diagram illustrating a backplane incorporating a pair of power supply units according to a preferred embodiment of the present invention; and
- [0017] Figure 2 is a more detailed schematic view of the power supply unit and controller according to a preferred embodiment of the invention.

Detailed Description

- [0018] An improved Power Supply Unit (PSU) 100 according to a preferred embodiment of the invention is shown in Figure 2. The PSU 100 incorporates a conventional type PSU 10" of the type adapted to fit inside a rack enclosure (not shown) and to transform an AC mains supply or a 48V DC supply (not shown) voltage to one or more DC supply rail levels – in the present example 3.3V, 5V and 12V. Typical examples of such PSU"s are manufactured by Artesyn and Celestica. Within the rack, the PSU 100 connects to a backplane 10 and supplies the DC levels through respective unidirectional devices 12, such as diodes, to corresponding tracks (not shown) running across the backplane 10 – the diodes enable more than one PSU to provide power to the same supply rail on the backplane. Other tracks on the backplane form one or more busses which interconnect other devices connected to the backplane. A more detailed description of a rack enclosure housing such power supply units is provided in relation to Figure 1 in the related

application entitled "Data Gathering Device for a Rack Enclosure" naming Aedan Diarmid Cailean Coffey et al as inventors (Attorney Docket No. PI29273).

[0019] The preferred embodiment is described in terms of a PSU for a rack enclosure housing a plurality of SCSI bus compatible devices, for example, a plurality of disk drives 80 comprising a RAID array system. These are in turn controlled from a host application 1 running on another device connected to the SCSI bus. The invention is nonetheless not limited to SCSI devices with other examples including Fibre Channel based devices.

[0020] According to the present invention, the PSU 100 incorporates a controller board 20. The board 20 includes a microcontroller 22 which in the preferred embodiment, is chosen from the NEC K0S family (uPD78F9177YGB-8ES). The PSU controller 22 gathers environment information from the PSU 10" and reports it, using, for example, a serial bus, to an Environment Monitoring System, which in the preferred embodiment is a processor 4 supporting the SCSI Environmental Services (SES) protocol. The SES processor 4 in turn communicates with a host application 1, for example, Vision supplied by Eurologic Systems Limited enabling user control and monitoring of environmental information. As described in the related application entitled "Data Gathering Device for a Rack Enclosure", it is possible, for redundancy reasons, to have more than one SES Processor 4, 4", however, for simplicity the present description will make reference to only one processor.

[0021] In the preferred embodiment, the PSU controller 22, rather than drawing its power supply Vcc from the closest available source, i.e. the PSU 10, instead draws its power from an appropriate one of the 3.3V, 5V or 12V supply rails on the backplane 10. This could be seen as adding a potential point of failure to the system, however, it means that should the PSU incorporating the PSU controller either completely fail or should its Vcc supply rail being used by the PSU controller fail, it is still possible for the PSU controller to draw power from any remaining operational PSU"s located in the enclosure and supplying power to the backplane 10.

[0022] So, while the addition of a microcontroller per se to the power supply could on the one hand be seen as an additional potential point of failure, the PSU controller of the preferred embodiment can remain operational and provide useful information to an operator long after its corresponding PSU has failed. As will be seen from the following description describing the operation of the controller, because failure or even potential failure is reported in software, failure messages indicate to an operator the reason for failure without requiring the operator to carry out any investigation of the hardware.

[0023] Thus, where a conventional PSU without a controller either partially or completely fails and indicates failure through the switching on or off of a LED or buzzer, the operator may not detect such a failure until, for example, catastrophic power failure when all rack PSUs fail. If on the other hand, the operator does notice the failure of a single PSU, the operator may then need to carry out exhaustive investigation to determine the fault, which may be as simple as an AC supply failure.

[0024] Using the PSU of the preferred embodiment, precise reasons for partial and/or complete failure of a PSU can be reported immediately and continuously to an operator even at a remote location, so preventing catastrophic failure of a system and improving maintainability of the system.

[0025] Turning now to the functionality of the PSU, the PSU controller 22 supports:

[0026] a set of analog to digital converters, ADC, which are used by an internal ADC for monitoring respective voltage and current levels of the 3.3V, 5V and 12V power rails in the PSU;

[0027] the serial bus, I²C, which is used to communicate to the SES Processor 4;

[0028] further digital inputs for general purpose I/O (GPIO); and

[0029] one LED output.

[0030] The complete I/O functionality for the controller of the preferred embodiment is fully described in Table 2.

Name	Type	Purpose
SMB_A0, SMB_A1	2 digital inputs	These are used to configure the 2 wire serial address for the microcontroller
Yellow LED	1 digital LED output	Used to control the Fault LED
3.3V, 5V, 12V Monitor IC 3.3V, 5V, 12, Imonitor IC	6 analog to digital converter (ADC) inputs	These are used to measure the current and the voltages on the 3.3, 5 and 12 volt lines
AC FAIL / AC SHUTDOWN	1 digital interrupt pin	Used to detect AC fail conditions
SMB_SCL, SMB_SDA	2 dedicated digital I/O lines with serial hardware	Used for the 2 wire I ² C serial bus interface
Temp Fail	1 digital input/interrupt	Used to detect a PSU temperature failure
3.3V, 5V, 12V Fail 3.3V, 5V, 12, IFail	6 digital inputs	Used to detect current and voltage failure on the 3.3, 5, and 12 v lines
Vref	1 analog reference voltage	Used to determine the voltages and currents measured on the ADC.
SCL_MSK, SDA_MSK	2 digital I/O lines	Used to talk to an EEPROM if a Mask part is used. The EEPROM stores the Part No. Serial No. and the PSU revision.

Table2

[0031] Using pins SMB_A0, SMB_A1 connected to the backplane 10, the PSU controller 22 supports four serial bus addresses, from 20 hex to 23 hex inclusive. These pins cause the address to be different based on the position of the power supply in the system, e.g. left = 20 hex and right = 21 hex. Thus, any rack enclosure can incorporate up to 4 PSUs individually addressable from one or more SES

processors.

[0032] A plurality of temperature sensors 26 (one corresponding to each PSU and located on the PSU controller board) are also connected to the 2 wire serial bus, but are not under control of the PSU controller 22. The temperature sensors addresses are of the form 010011XXb where XX are the same as the 2 least significant bits in the PSU serial address, thus the temperature sensors have addresses from 4C hex to 4F hex inclusive.

[0033] i.e. PSU = 20 hex Temperature Sensor = 4C hex

[0034] and PSU = 22 hex Temperature Sensor = 4E hex.

[0035] Turning now to the analog inputs, the voltage and current values reported are each an 8 bit (1 byte) value read from the internal ADC, where 00 hex is 0V and FF hex is the maximum voltage that can be measured on that voltage rail. (Where there is a requirement to do some calculations on this data it is carried out in the SES Processor microcontroller using a scaling factor/resolution, explained below, associated with the analog signal being measured.)

[0036] The ADC voltage range is between Gnd and Vcc of the microcontroller. For the PSU controller the analog voltages are referred to in the present description as AGnd and AVcc.

[0037] $Gnd = 0V = AGND$ and $Vcc = 5V(+/-0.3V) = AVcc$.

[0038] Where the maximum voltage for a PSU 10" that can be measured on the 3.3V rail is, say 5V, the bit resolution or scaling factor is 19.5mV. The 5V and 12V rails, on the other hand, need to be scaled down to a range between AGND (Gnd) and AVCC ($Vcc +/- 0.3V$) in order to measure their voltage levels.

[0039] The ADC is preferably capable of measuring up to 6.5V maximum on the 5V line, so the bit resolution is 24.5mV. The ADC is further preferably capable of measuring up to 15V maximum on the 12V line, so the bit resolution is 58.6mV.

[0040] E.g. On 5V rail actual voltage = 4.85V

[0041] ADC reading = 110001010b = C5 hex

[0042] PSU Voltage measurement =

[0043] (ADC reading) * (bit resolution) =

[0044] C5 hex * (0.0245) = 4.8265V

[0045] The PSU controller also monitors the current on the 3.3V, 5V and 12V power rails in the PSU. The current is calculated using the scaling values according to the PSU manufacturer for translating a measured voltage level to a current value. The 3.3V, 5V and 12V rails are divided down to a range between AGND and AVCC so they can be measured by the ADC.

[0046] Where the maximum current capable of being measured on the 3.3V line, is say 0.3A, the bit resolution is 1.17mA. Where the maximum current capable of measured on the 5V line is, say 35A, the bit resolution is 136.7mA. Where the maximum current capable of measured on the 12V line is, say 50A, the bit resolution is 195mA.

[0047] E.g. On 5V rail actual current = 27A

[0048] ADC reading = 110001010b = C5 hex

[0049] PSU Current measurement =

[0050] (ADC reading) * (bit resolution) =

[0051] C5 hex * (0.1367) = 26.9299A

[0052] The PSU controller board of the preferred embodiment includes 34 bytes of permanent data storage. The information that these bytes contain includes: the serial number, part number, and the scaling factors for voltage and current conversion (according to the PSU manufacturer).

[0053] This information is stored in flash code space within the PSU controller at a fixed address. Alternatively, where PSU controller code has settled down, a microcontroller mask part can be introduced to reduce design cost. This mask

part, however, does not allow changes to permanent data in manufacturing, and so to allow for this, an EEPROM 28 is included in the design. (This component is not stuffed in the manufactured product with the flash microcontroller.) Where the data specific to the PSU and setup during manufacturing is stored in the EEPROM, it can be loaded using programmable logic tools from for example Integrated Circuit Technology Corp. or another two wire serial master. (This would also require the addition of new commands, to allow the scaling factors, part number, and serial number to be programmed over the serial bus connecting the SES Processor to the PSU controller.)

[0054] Programming the PSU controller comprises:

[0055] 1. Once only, compiling the PSU controller code including the firmware revision and the predefined scaling values for the voltage and current calculations generating a hex or binary file for download.

[0056] 2. For every PSU, generating in production: the PSU serial number, part number and version number.

[0057] 3. For every PSU, running a program to integrate the PSU serial number, part number and version number in to the hex or binary file for download.

[0058] 4. For every PSU, running a download utility to program the PSU microcontroller.

[0059] As indicated above, communication between the SES Processor and the PSU controller is carried out using a serial bus. The present embodiment is described in terms of an I²C bus, where the PSU controller supports a serial slave mode only. Alternative busses include: RS-232, Serial Peripheral Interface (SPI) or an Enclosure Services Interface (ESI) bus.

[0060] When reading data from the PSU controller, the PSU controller is addressed as defined by the I²C bus at the correct serial address – valid addresses being from 20 hex to 23 hex. The serial read/write bit in the address byte which is set to read is then transmitted by SES Processor. The PSU controller device then acknowledges

the receipt of its address, launches an internal interrupt service routine to process the read request and stretches the clock until it has processed the interrupt involving setting up the data to transmit.

[0061] The SES controller then toggles the serial bus clock 9 times (8 bits of data and 1 acknowledge) for each byte it wants to receive from the PSU controller. The PSU controller data is transmitted starting with byte 0 followed by byte 1 and continues until the SES Processor stops the transfer. If the number of bytes being clocked out exceeds the page length the byte 0xBC hex will be transmitted.

[0062]

The PSU controller is capable of receiving one or more bytes of data and transmitting 1 to 56 bytes of data, and the possible 56 bytes of data are shown in Table 1. This data is preferably held within the PSU controller in SRAM 30 or general purpose working registers (not shown). The first two bytes is read/write data, and the others are read only. When reading from the PSU controller, all 56 bytes can be read in the order shown. If more data bytes are written by the SES Processor than the single byte supported, then only the first byte is treated as valid the extra bytes received are ignored. If more than the maximum number of bytes available are requested by the SES Processor, the PSU microcontroller outputs FF hex after the last valid data byte.

Byte	Direction	Name	Description
Byte 0	R/W	PSU Summary 0	Power Supply Summary 0
Byte 1	R/W	PSU Summary 1	Power Supply Summary 1
Byte 2	R	PSU Status 0	Power Supply Status 0
Byte 3	R	Reserved	Reserved
Byte 4	R	3.3V Voltage Reading	Actual voltage on the 3.3V line
Byte 5	R	5V Voltage Reading	Actual voltage on the 5V line
Byte 6	R	12V Voltage Reading	Actual voltage on the 12V line
Byte 7	R	3.3V Current Reading	Actual current on the 3.3V line
Byte 8	R	5V Current Reading	Actual current on the 5V line
Byte 9	R	12V Current Reading	Actual current on the 12V line
Byte 10,11**	R	3.3V Resolution	3.3V Resolution (19.5mV)
Byte 12,13**	R	5V Resolution	5V Resolution (24.5mV)
Byte 14,15**	R	12V Resolution	12V Resolution (58.6mV)
Byte 16,17**	R	3.3A Resolution	3.3V current Resolution (1.17mA)
Byte 18,19**	R	5A Resolution	5V current Resolution (136.7mA)
Byte 20,21**	R	12A Resolution	12V current Resolution (195mA)
Byte 22**	R	PSU Part Number	PSU part number also on label of PSU
..			
Byte 37			
Byte 38*	R	PSU Serial Number	PSU serial number also on bar code label
..			
Byte 53			
Byte 54	R	Build Number	A number directly related to the revision of the firmware
Byte 55			

* Permanent data stored in Flash from manufacturing

** Permanent data from Firmware build

Table1

[0063] Turning now to the details of the information in Table 1, *PSU Summary*, which is stored in a register, contains summary information on the current status of the PSU. Using these two bytes, it is possible to confirm correct operation of the PSU by a single read of this register. The two bytes are divided into the following bits as shown in Table 3.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Led Status and Request			Auto	AC Fail Latched	AC Fail	PSU Shutdown	Summary
Rsvd	Rsvd	Rsvd	Rsvd	Serial Comms Timeout	Power Up Ack	Temp Fail Latched	Temp Fail

Table 3

- [0064] *Summary* is Set to a 1 when a fault is detected in the PSU, and is set to a 0 if the power supply is operating normally. Writing to this bit has no effect.
- [0065] *PSU Shutdown* is set to a 1 when the power supply is forced off or non-operational due to an internal problem, 0 otherwise. Writing to this bit has no effect. The power supply hardware reports this state to the PSU controller via the AC Fail/Shutdown signal. When the shutdown signal causes an interrupt, the PSU ceases to operate and requires a power reset to recover. The PSU controller records all the digital inputs when this situation occurs and the PSU controller stops recording new values on the ADC channels. When accessed, the PSU controller only reports these latched values. After power on/off cycling of the PSU controller, the PSU controller returns to normal operation.
- [0066] *AC Fail* is set to a 1 when the AC input voltage is below approximately 75VAC, 0 otherwise. Writing to this bit has no effect.
- [0067] *AC Fail Latched* records any AC fail until the SES Processor has seen it and requests that it be cleared. It is set to 1 if an AC_FAIL has been detected and recovered, and is recorded using an edge triggered interrupt. In the preferred embodiment, this bit can only be reset by the SES Processor, by writing a "1" to clear this bit.
- [0068] *Auto* is set to a 0 when the PSU controller is working in autonomous mode, otherwise set to 1 by the SES Processor. The PSU controller, by default, has this bit cleared, as this allows it to report the PSU status to an operator via the LED. If the SES Processor sets this bit, the PSU controller does not report the PSU status via the LED – rather the status is under the control of the SES Processor. Once the PSU is under control of the SES Processor, the only way to reset it to autonomous mode is

to reset the PSU controller or write the Auto bit back to 0.

- [0069] In relation to *Led Status and Request* , the LED flashes on and off at different rate depending on the pattern set in these 3 bits. The flash rates are defined in Table 4 below.

Priority	LED Pattern	Description
0x00	0x0000	LED permanently off
0x01	0xaaaa	LED on for 125mS LED off for 125 mS - repeat forever
.....		
0x06	0x0f0f	LED on for 500mS LED off for 500 mS - repeat forever
0x07	0xffff	LED permanently on

Table 4

- [0070] *Temp Fail* is set to 1 if the power supply is in an over temperature condition, 0 otherwise.
- [0071] *Temp Fail Latched* records any Temp fail until the SES Processor has seen it and requests that it be cleared. It is set to 1 if a Temp fail has been detected and gone away, and is recorded using an edge triggered interrupt. Preferably, this bit can only be reset by the SES Processor, by writing a 1 to clear this bit.
- [0072] *Power Up Ack* is set to "1" every time the PSU controller is powered up. Writing a "1" to this location clears this bit.
- [0073] *Serial Comms Timeout* is set when the PSU controller board detects a timeout on the serial bus. This normally occurs if the Master device (the SES Processor) does not supply a clock signal within 100mS. A 1 must be written to this bit to set it back to 0.
- [0074] *PSU Status* , is again stored in a register and contains information on the current status of the PSU. This is read from 6 "Dig I/O" inputs supplied by the PSU. (All the calculations for determining that a voltage and current is outside predetermined limits are carried out by the PSU hardware.) These signals show a failure when the power supply has gone outside the predefined parameters. The

bits in this register are defined as shown in Table 5 below – all those marked reserved return zero when read. The remainder are each Set to 1 if the corresponding output voltage/current is outside predetermined limits, 0 otherwise. All writes to this register are ignored.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Rsvd	Rsvd	12A	12v	5A	5v	3.3A	3.3v
		Fail	Fail	Fail	Fail	Fail	Fail

Table 5

[0075] *3.3V, 5V, 12V Voltage Reading* – when the corresponding register is read, it returns a hex value corresponding to the analog voltage measured on the 3.3V, 5V or 12V output. The voltage range is from 00 to FF hex where FF hex will be full scale of the ADC, i.e. the maximum measurable voltage. The analog voltage range of the ADC is from 0 to 5, 0 to 6.5 and 0 to 15 volts respectively and the resolution is in steps of 19.5mV/bit, 25.4mV/bit and 58.6mV/bit. Writes to these registers are ignored.

[0076] The SES Processor in turn uses the following method to calculate the voltage read by the PSU controller.

[0077] (Voltage measurement) * (Resolution)

[0078] For example if 7C hex is read in from the PSU in the PSU controller 3.3V Register

[0079] 7C hex = 124 decimal => (124)*(0.0195) = 2.418V.

[0080] *3.3V, 5V, 12V Current Reading* – these registers return a value corresponding to the analog current sourced from the corresponding voltage rail output. The analog current ranges are from 0A to .3A, 0A to 35A and 0A to 50A respectively and the resolution of the ADC is in steps of 1.17mA/bit, 136.7Ma/bit and 195mA/bit. Writes to these registers are ignored. These voltages are scaled using PSU manufacturer scaling factors in order to calculate the associated current. The current range is from 00 to FF hex where FF hex is full scale of the ADC, i.e. the maximum measurable current.

[0081] The SES Processor uses the following method to calculate the current read by the PSU controller.

[0082] (Current measurement) * (Resolution)

[0083] For example if 7C hex is read in from the PSU in the PSU controller 3.3A Register

[0084] 7C hex = 124 decimal => (124)*(0.00117) = 0.14508A.

[0085] *3.3V, 5V, 12V Resolution* – these values are used by the SES Processor to calculate the actual voltage measured on the respective voltage lines by the PSU controller. These values are fixed in the firmware and built in to the code at compile time as explained above. In the present example, these values are as shown below:

VOLTAGE SCALING FACTORS		
Power Rail	Resolution	Register Value
3.3 Volt	19.5mV/bit	279Eh
5 Volt	25.4mV/bit	29ECh
12 Volt	58.6mV/bit	124Ah

[0086] Data written to these locations is ignored. The resolution is stored in two bytes using a dedicated 2 byte floating point method defined below.

[0087] *3.3V, 5V, 12V Current Resolution* – these values are used by the SES Processor to calculate the actual current measured on the respective voltage lines by the PSU controller. These values are fixed in the firmware and built in to the code at compile time as explained above. In the present example, these values are as shown below:

CURRENT SCALING FACTORS		
Power Rail	Resolution	Register Value
3.3V Current	1.17mA/bit	3492h
5V Current	136.7mA/bit	1557h
12V Current	195mA/bit	179Eh

[0088] Data written to these locations is ignored and the resolution is stored in two bytes in the 2 byte floating point method mentioned above. In this

regard, the NEC microcontroller (as with some other controllers) normally represents floating point values with a 4 byte number in standard IEEE format. This format is very long for the present application as there is a requirement to store the 6 floating point resolution values. It should also be noted that the PSU controller does not have to carry out any mathematical operations on these numbers, so the structure need not be in a standard format. For these reasons, in the preferred embodiment, a 2-byte format is used.

[0089] The basis to this floating point notation is that the scaling factors for the power supply can range from millivolts and milliamps to Volts and Amps. The Mantissa is defined as the scaling factor in thousands of millivolts or milliamps. The Exponent is the number of decimal point shifts required to evaluate the correct scaling factor.

[0090] The 16 bits are divided into two bit sections, 12 bits and 4 bits. The first 12 bits represent the Mantissa, a decimal millivolt number in hex format; while the 4 bits represent the exponent value ($\times 10^{-\text{value}}$). This is shown in a diagram below and a few examples are shown. The range of this notation is from 4.095×10^{-12} mV = (FFFF hex) to $4096 \text{ mV} = 4.096 \text{ V}$ (0FFF hex). This notation requires that the first byte of the Mantissa always has a hex value in it. If the converted decimal value exceeds 3 hex digits then the decimal value must be reduced by a factor of 10, i.e. $58.6 \text{ mV} = 5860 \times 10^{-2} \text{ mV} = 586 \times 10^{-1}$. This gives a scaling factor of 124A hex.

15 12	11 0
Exponent	Mantissa

[0091] e.g.

[0092] $19.5 \text{ mV} = (\text{decimal}) 1950 \times 10^{-2} \text{ mV} = 79 \text{ E} \times 10^{-2} \text{ mV} = 279 \text{ E hex}$

[0093] $1.17 \text{ mA} = (\text{decimal}) 1170 \times 10^{-3} \text{ mV} = 492 \times 10^{-3} \text{ mV} = 3492 \text{ hex}$

[0094]

It will be seen that by building the scaling factors into the PSU controller

firmware, upper level software may then be independent of the PSU used as this simply needs to retrieve a measured value and corresponding scaling factor to determine the voltage being supplied by the PSU.

- [0095] *PSU Part Number* – this data is permanent and programmed in production as explained above. When these registers are read, they return the PSU part number as a string of up to 16 hexadecimal characters. The part number is made up of the firmware revision (FM), the hardware revision (HM), the system type and the hardware vendor as shown in table 6. (Each byte of HM and FM are valid from 00 to 99 ASCII) Preferably, the string matches the part number printed on the PSU label. Data written to this register is ignored.

Byte	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	Reserved					Hardware Vendor	System	HM	HM	FM	FM					

Table 6

- [0096] *PSU Serial Number* – this data is permanent and programmed in production as explained above. When these registers are read they return the PSU serial number as a string of up to 16 ASCII characters. The serial number string should match the serial number printed on the PSU label and is the responsibility of operations. Data written to this register is ignored.

- [0097] *Build Number* is a two byte hexadecimal value that starts at 0x0001 and is incremented every time a new release of the PSU controller firmware issued. This number is not a revision number but using this number and a firmware source code database one can build any revision of the PSU controller firmware.

[0098]

On the other hand, when writing to the PSU controller, the first byte is a command byte and the subsequent bytes are data that are to be operated upon. In the preferred embodiment, data that is written to the PSU controller should be limited to 17 bytes, the first byte will be a command byte and the next 16 will be the data or payload. The commands that the PSU controller supports are listed in Table 7 with a data length and description.

Command	Command Byte	Data Length	Data Description
Status	0x01	2 bytes	Status Data to be implemented by the PSU controller
Error Priority	0x02	11 bytes	The error led information for each error in the PSU controller <i>Note 1</i>
Led Pattern	0x03	8 integers (16 bytes)	The pattern that each led priority is allocated <i>Note 1</i>
Scaling Factor	0x04	8 integers (16 bytes)	The scaling factor associated with each analog measurement <i>Note 1</i>
Part Number	0x05	16 bytes	The PSU controller board part number <i>Note 1</i>
Serial Number	0x06	16 bytes	The PSU controller board serial Number <i>Note 1</i>
Reboot	0xbb	0 bytes	This will cause the PSU controller to restart

Note 1 – Without an EEPROM, changes to this information are received and stored in RAM, so everytime the PSU controller is reset this information has to be updated. With an EEPROM, this information is written out to the EEPROM and stored permanently

Table 7

[0099] *Status Command* is followed by 2 bytes of information, with all other bytes being ignored. The information in these bytes is as shown in Table 8 and valid as described in relation to PSU Summary above.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Led Status and Request		Enable	Reset AC	Rsud	Rsud	Rsud	Rsud
		Auto Mode	Fail Latched				
Rsud	Rsud	Rsud	Rsud	Rsud	Power Up Ack	Temp Fail Latched	Rsud

Table 8

[0100]

Error Priority – the PSU controller supports 11 possible errors, each of these is assigned a priority. The possible priorities are from 0 to 7 with 0 being the lowest. By default all errors are assigned a priority of 7. This priority is used to select a LED pattern to be displayed when the error occurs. If 2 errors are detected in the system the error with the highest priority will be used to select the led pattern to be displayed. The table below shows the possible errors and the order that the data should be presented to the PSU controller.

Failure	Default Priority
AC fail	0x07
Temp fail	0x07
3.3V fail	0x07
5V fail	0x07
12V fail	0x07
3.3I fail	0x07
5I fail	0x07
12I fail	0x07
PSU Shutdown	0x07
AC fail latch	0x07
Temp fail latch	0x07

[0101] *Led Pattern* – as explained above, the PSU controller supports 8 possible LED Patterns, and these are directly related to the priority of an error. So using Table 4 above 0x00 hex = 0x0000 hex will force the LED to be off all the time and 0x07 hex = 0xffff hex will force the LED to be on all the time.

[0102] *Scaling Factor* – as explained above, the PSU controller supports 6 analog readings, each of which have an associated scaling factor or resolution. The ADC on the PSU controller only measures from 0 to 5 volts so the analog signals must be scaled to represent the true value in the power supply. These values are fixed based on the engineering specification for each PSU 10. The values also take the floating point format described below. The order that the PSU controller expects to receive this data is shown below.

Resolution for 3.3V
Resolution for 5V
Resolution for 12V
Resolution for 3.3V Current
Resolution for 5V Current
Resolution for 12V Current

[0103] *Part Number* – the part number for the PSU controller is 16 bytes long and the format used is defined above.

[0104] *Serial Number* – the serial number for the PSU controller is also 16 bytes long

and the format used is defined above.

- [0105] *Reboot* causes the PSU controller to restart from the beginning of its code. This reboot appears to the user and to the SES cessor as a power-on reset of the PSU controller. The PSU power up bit is set after this command is executed, any data in RAM wi be lost after this command is executed.

09531645 0951604
T06T00 549T8920